



Πίνακες και Δομές δεδομένων



Μονοδιάστατοι πίνακες

Στο Α΄ Τεύχος μελετήσαμε πώς λύνουμε αλγοριθμικά κάποια προβλήματα, αποθηκεύοντας δεδομένα σε μεταβλητές και εφαρμόζοντας σε αυτές τις τρεις βασικές αλγοριθμικές δομές (δομή ακολουθίας, δομή επιλογής και δομή επανάληψης). Μπορούμε άραγε να αντιμετωπίσουμε αποτελεσματικά οποιοδήποτε πρόβλημα χρησιμοποιώντας ως δομές αποθήκευσης τιμών μόνο μεταβλητές; Προκαταβολικά, η απάντηση είναι αρνητική. Στο παρόν και στα επόμενα 5 κεφάλαια θα εξετάσουμε νέες δομές αποθήκευσης δεδομένων, τους πίνακες, οι οποίοι θα μας επιτρέψουν να λύσουμε με κομψό και σχετικά εύκολο τρόπο μια μεγάλη κατηγορία προβλημάτων.

Ένα παράδειγμα στο οποίο η χρήση απλά και μόνο μεταβλητών είναι προβληματική δίνεται στη συνέχεια.

17.1 Να γίνει αλγόριθμος ο οποίος:

- θα διαβάζει το ύψος 100 μαθητών και θα εμφανίζει το μέσο ύψος τους.
- θα εμφανίζει το πλήθος των μαθητών που έχουν ύψος μεγαλύτερο του μέσου ύψους.

Απάντηση

Το (α) ερώτημα το έχουμε ήδη αντιμετωπίσει σε διάφορες παραλλαγές του σε προηγούμενα κεφάλαια (δες, για παράδειγμα, το θέμα 14.2). Συγκεκριμένα το κατάλληλο τμήμα αλγόριθμου είναι το ακόλουθο:

$\Sigma \leftarrow 0$

για μαθητή από 1 μέχρι 100

 διάβασε ύψος

$\Sigma \leftarrow \Sigma + \text{ύψος}$

τέλος_επανάληψης

$MO \leftarrow \Sigma / 100$

εμφάνισε MO

Αν κληθούμε τώρα να απαντήσουμε στο (β) ερώτημα, τότε τα πράγματα γίνονται δύσκολα. Αφού υπολογίστηκε ο μέσος όρος, θα πρέπει να συγκρίνουμε όλα τα ύψη

με αυτόν. Όμως στη μεταβλητή ύψος βρίσκεται πλέον αποθηκευμένο μόνο το τελευταίο ύψος. Τα υπόλοιπα 99 χάθηκαν. Έτσι, ή πρέπει να μας τα ξαναδώσει ο χρήστης (πράγμα που δεν θα το ήθελε καθόλου) ή θα έπρεπε εξαρχής να τα είχαμε αποθηκεύσει σε 100 διαφορετικές μεταβλητές (πράγμα πολύ δύσκολο, αφού θα έπρεπε να γράψουμε *διάβασε ύψος1, ύψος2, ύψος3, ..., ύψος100* χωρίς βέβαια τα ομοιωματικά). Εδώ, λοιπόν, απαιτούνται νέες δομές αποθήκευσης τιμών (δομές δεδομένων), μία εκ των οποίων είναι ο μονοδιάστατος πίνακας που θα δούμε στη συνέχεια.

Εισαγωγικά στοιχεία πινάκων - μονοδιάστατων πινάκων

17.2 ΘΕΩΡΙΑ Τι ορίζουμε ως πίνακα;

Απάντηση

Πίνακας είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Καθένα από τα αντικείμενα που απαρτίζουν τον πίνακα λέγεται στοιχείο του πίνακα. Η αναφορά στα στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από έναν δείκτη.

17.3 ΘΕΩΡΙΑ Τι είναι ένας μονοδιάστατος πίνακας;

Απάντηση

Ένας μονοδιάστατος πίνακας είναι μια δομή δεδομένων που μας επιτρέπει στο όνομα μιας μεταβλητής να καταχωρίσουμε περισσότερα από ένα στοιχεία του ίδιου τύπου. Καθένα από αυτά τα στοιχεία διαθέτει έναν μοναδικό δείκτη, ο οποίος προσδιορίζει τη θέση του μέσα στον πίνακα.

Για να κατανοήσουμε την έννοια του μονοδιάστατου πίνακα αλλά και γενικότερα της δομής πίνακα, μπορούμε να τον φανταστούμε, σε αναλογία με τις μεταβλητές, ως ένα σύνολο από θυρίδες οι οποίες έχουν ένα κοινό όνομα που σε καθεμία από αυτές αντιστοιχεί ένας μοναδικός δείκτης. Σε κάθε θυρίδα μπορεί να αποθηκευτεί μία τιμή, η οποία πρέπει να είναι του ίδιου τύπου με τις τιμές των υπόλοιπων θυρίδων.



17.4 ΘΕΩΡΙΑ Ποια είναι τα χαρακτηριστικά ενός μονοδιάστατου πίνακα;

Απάντηση

Τα χαρακτηριστικά ενός μονοδιάστατου πίνακα είναι τα ακόλουθα:

► Όνομα πίνακα

Το όνομα ενός πίνακα ακολουθεί τους ίδιους κανόνες ονοματολογίας με τις μεταβλητές.

► Μέγεθος πίνακα

Το μέγεθος ενός μονοδιάστατου πίνακα είναι ένας ακέραιος θετικός αριθμός που παραμένει σταθερός κατά τη διάρκεια εκτέλεσης του προγράμματος ή του αλγόριθμου και προσδιορίζει το πλήθος των θέσεων του πίνακα.

► Τύπος δεδομένων ενός πίνακα

Ο τύπος δεδομένων των στοιχείων ενός πίνακα μπορεί να είναι ακέραιος, πραγματικός, λογικός ή χαρακτήρας. Όλα τα στοιχεία του πρέπει να είναι του ίδιου τύπου.

17.5

ΘΕΩΡΙΑ

Πώς δηλώνουμε έναν μονοδιάστατο πίνακα σε ένα πρόγραμμα;

Απάντηση

Στο τμήμα δήλωσης του προγράμματος καθορίζονται τα τρία χαρακτηριστικά των μονοδιάστατων πινάκων που είδαμε στο προηγούμενο θέμα ως εξής:

τύπος: όνομα_πίνακα[μέγεθος]

Παραδείγματα:

ακέραιες: βαθμός[5] – Δηλώνουμε έναν πίνακα 5 θέσεων (στοιχείων) με όνομα βαθμός, όπου σε καθεμία από τις θέσεις του μπορούμε να καταχωρίσουμε ακέραιες τιμές.

χαρακτήρες: όνομα_πελάτη[1000] – Δηλώνουμε έναν πίνακα 1.000 θέσεων, με όνομα *όνομα_πελάτη*, όπου σε καθεμία από αυτές μπορούμε να καταχωρίσουμε κάποια σειρά χαρακτήρων.

λογικές: A[10], B[30], x – Δηλώνουμε έναν πίνακα A με 10 θέσεις, έναν πίνακα B με 30 θέσεις, στις οποίες θα καταχωριστούν λογικές τιμές (Αληθής ή Ψευδής), καθώς και μια μεταβλητή με όνομα x λογικού τύπου.

Προσοχή

► Δεν πρέπει να ξεχνάμε ότι, όταν δηλώνουμε μια μεταβλητή, δεσμεύουμε μέχρι την ολοκλήρωση του προγράμματός μας ένα πολύ μικρό μέρος της μνήμης του υπολογιστή, μέσα στο οποίο θα καταχωριστεί το περιεχόμενο της μεταβλητής. Αντίστοιχα, όταν δηλώνουμε έναν πίνακα, δεσμεύουμε τόσες θέσεις της μνήμης του υπολογιστή όσες δηλώνει ο αριθμός μέσα στις αγκύλες κατά τη δήλωση του πίνακα. Μάλιστα αυτές οι θέσεις θα είναι συνεχόμενες.

Για παράδειγμα: **χαρακτήρες: όνομα_πελάτη[1000]** – Δεσμεύουμε 1.000 συνεχόμενες θέσεις μνήμης, ώστε να καταχωριστούν κατά την εκτέλεση του προγράμματος 1.000 ονόματα.

- ▶ Όλα τα στοιχεία του πίνακα είναι του ίδιου τύπου, δηλαδή δεν μπορεί να δηλώσω για παράδειγμα έναν πίνακα που κάποιες θέσεις να περιέχουν χαρακτήρες και κάποιες άλλες αριθμούς.
- ▶ Το μέγεθος του πίνακα προσδιορίζεται στο τμήμα δηλώσεων και δεν αλλάζει κατά τη φάση εκτέλεσης του προγράμματος.

17.6 Πώς γίνεται η αναφορά σε ένα στοιχείο ενός μονοδιάστατου πίνακα;

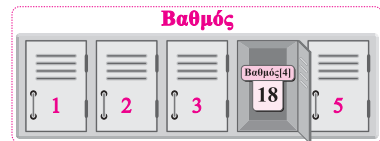
Απάντηση

Οποιαδήποτε αναφορά σε έναν μονοδιάστατο πίνακα γίνεται σε καθένα από τα στοιχεία του ξεχωριστά και ποτέ συνολικά σε ολόκληρο τον πίνακα. Συγκεκριμένα, για να αναφερθούμε σε κάποιο από τα στοιχεία ενός πίνακα, χρησιμοποιούμε το όνομα του πίνακα συνοδευόμενο από τον δείκτη του στοιχείου κλεισμένο σε αγκύλες:

όνομα_πίνακα[i] Αναφερόμαστε στο στοιχείο που βρίσκεται στην i-οστή θέση του πίνακα.

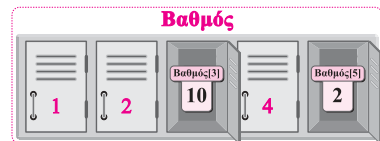
Παραδείγματα:

α) εμφάνισε βαθμός[4] – Εμφανίζει στην οθόνη το περιεχόμενο της τέταρτης θέσης του πίνακα *βαθμός*, δηλαδή για τον διπλανό πίνακα εμφανίζει το 18.



β) κ ← 5

$x \leftarrow A[3] * A[5]$ – Καταχωρίζεται στη μεταβλητή *x* το γινόμενο των τιμών που βρίσκονται στις θέσεις 3 και 5 του πίνακα *A*, δηλαδή των τιμών 10 και 2.



17.7 Πώς μπορούμε να δώσουμε τιμές σε έναν μονοδιάστατο πίνακα;

Απάντηση

Οι τιμές σε έναν πίνακα θα καταχωρίζονται σε κάθε θέση του ξεχωριστά και όχι σε ολόκληρο τον πίνακα. Όπως στις μεταβλητές, έτσι και στους πίνακες υπάρχουν δύο τρόποι για να καταχωρίσουμε τιμές. Ένας με τη βοήθεια της εντολής **διάβασε** και ένας με τη βοήθεια της εντολής εκχώρησης τιμής.

Ας δούμε μερικά παραδείγματα εντολών εισαγωγής τιμών σε έναν πίνακα *A* που έχει δηλωθεί ως τύπου πραγματικός (**πραγματικές**: $A[4]$). Το σχόλιο δεξιά της εντολής περιγράφει τι κάνει ο υπολογιστής εκτελώντας την αντίστοιχη εντολή.

$A[3] \leftarrow 14$

! Η τιμή 14 καταχωρίζεται στη θέση 3 του πίνακα.

$x \leftarrow 1$

! Το 1 καταχωρίζεται ως περιεχόμενο στη μεταβλητή *x*.

$A[x] \leftarrow 3$! Το 3 καταχωρίζεται στη θέση που ορίζει το περιεχόμενο του x (δηλαδή στη θέση 1).
$A[x + 1] \leftarrow A[x] + 1$! Το αποτέλεσμα της πρόσθεσης του 1 με το περιεχόμενο της θέσης του πίνακα A που ορίζει το x καταχωρίζεται στην επόμενη θέση από εκείνη που ορίζει το περιεχόμενο του x .
$A[4] \leftarrow A[x] + A[x + 1]$! Προστίθεται το περιεχόμενο του πίνακα που ορίζεται από το x με το περιεχόμενο της επόμενης θέσης και το αποτέλεσμα καταχωρίζεται στην 4η θέση του πίνακα.
διάβασε $A[x + 3]$! Ζητείται μια τιμή από τον χρήστη του προγράμματος η οποία καταχωρίζεται στη θέση $x + 3$ του πίνακα A .
$A[4] \leftarrow A[4] + 1$! Το περιεχόμενο της θέσης 4 του πίνακα A αυξάνεται κατά 1.

Παρατήρηση

Όπως είδαμε στα παραδείγματα των θεμάτων 17.6 και 17.7, η αναφορά σε ένα στοιχείο του πίνακα γίνεται με τη βοήθεια ενός δείκτη, ο οποίος μπορεί να είναι είτε ακέραιος αριθμός (για παράδειγμα $A[3]$) ή μεταβλητή τύπου ακεραίου (για παράδειγμα $A[x]$) ή τέλος κάποια μαθηματική έκφραση (για παράδειγμα $A[x + 1]$). Στην περίπτωση της μαθηματικής έκφρασης το αποτέλεσμά της πρέπει όχι μόνο να είναι ακέραιος αριθμός αλλά να είναι μεγαλύτερος ή ίσος της μονάδας και μικρότερος ή ίσος του μεγέθους του πίνακα.

17.8 Ποιο είναι το περιεχόμενο των στοιχείων ενός πίνακα πριν πάρουν τιμή με τη βοήθεια της εντολής **διάβασε** ή της εντολής εκχώρησης τιμής;

Απάντηση

Κάθε στοιχείο ενός πίνακα αρχικά περιέχει κάποια **απροσδιόριστη τιμή**, δηλαδή μια τυχαία τιμή. Έτσι, λοιπόν, στους αλγόριθμους και στα προγράμματα που θα υλοποιήσουμε όλα τα στοιχεία ενός πίνακα πριν χρησιμοποιηθούν σε μια έκφραση πρέπει να έχουν προηγουμένως λάβει τιμή με τη βοήθεια της εντολής **διάβασε** ή με τη βοήθεια της εντολής εκχώρησης.

17.9 Τι τιμές θα έχουν τα στοιχεία του πίνακα B έπειτα από την εκτέλεση καθεμίας από τις ακόλουθες εντολές;

$B[4] \leftarrow 0$

$x \leftarrow 1$

$B[x] \leftarrow B[4] + 6$

$B[x + 1] \leftarrow B[x] + 2$

$B[x + 2] \leftarrow B[x + 1]$

$B[5] \leftarrow B[4] + 2$

$B[5] \leftarrow B[5] + x$

Απάντηση

Εντολές	Πίνακας B	Μεταβλητή x											
! Αρχικές τιμές	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>?</td><td>?</td><td>?</td><td>?</td><td>?</td></tr></table>	1	2	3	4	5	?	?	?	?	?	<table border="1"><tr><td>?</td></tr></table>	?
1	2	3	4	5									
?	?	?	?	?									
?													
$B[4] \leftarrow 0$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>?</td><td>?</td><td>?</td><td>0</td><td>?</td></tr></table>	1	2	3	4	5	?	?	?	0	?	<table border="1"><tr><td>?</td></tr></table>	?
1	2	3	4	5									
?	?	?	0	?									
?													
$x \leftarrow 1$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>?</td><td>?</td><td>?</td><td>0</td><td>?</td></tr></table>	1	2	3	4	5	?	?	?	0	?	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
?	?	?	0	?									
1													
$B[x] \leftarrow B[4] + 6$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>?</td><td>?</td><td>0</td><td>?</td></tr></table>	1	2	3	4	5	6	?	?	0	?	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
6	?	?	0	?									
1													
$B[x + 1] \leftarrow B[x] + 2$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>8</td><td>?</td><td>0</td><td>?</td></tr></table>	1	2	3	4	5	6	8	?	0	?	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
6	8	?	0	?									
1													
$B[x + 2] \leftarrow B[x + 1]$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>8</td><td>8</td><td>0</td><td>?</td></tr></table>	1	2	3	4	5	6	8	8	0	?	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
6	8	8	0	?									
1													
$B[5] \leftarrow B[4] + 2$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>8</td><td>8</td><td>0</td><td>2</td></tr></table>	1	2	3	4	5	6	8	8	0	2	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
6	8	8	0	2									
1													
$B[5] \leftarrow B[5] + x$	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>8</td><td>8</td><td>0</td><td>3</td></tr></table>	1	2	3	4	5	6	8	8	0	3	<table border="1"><tr><td>1</td></tr></table>	1
1	2	3	4	5									
6	8	8	0	3									
1													

Προσοχή

Από τον παραπάνω πίνακα βλέπουμε ότι όλες οι θέσεις του αρχικά περιέχουν κάποια απροσδιόριστη τιμή, την οποία στο βιβλίο αυτό τη δηλώνουμε με το σύμβολο ?.

Ερώτηση

Τι θα συνέβαινε αν για τον παραπάνω πίνακα 5 θέσεων επιχειρούσαμε να καταχωρίσουμε στοιχείο σε θέση που δεν υπάρχει; Ποιο κριτήριο δεν ικανοποιείται;

Απάντηση

Το πρόγραμμα θα τερμάτιζε απότομα. Αν, για παράδειγμα, γράφαμε την εντολή $B[6] \leftarrow 10$, ο υπολογιστής θα προσπαθούσε να καταχωρίσει το 10 σε θέση του

πίνακα που δεν έχει δεσμευτεί κατάλληλος χώρος για αυτήν. Έτσι, λοιπόν, τη στιγμή εκτέλεσής της το πρόγραμμα θα τερμάτιζε απότομα. Το κριτήριο που δεν ικανοποιείται από ένα τέτοιο λάθος είναι αυτό της **καθοριστικότητας**.

Βασικές επεξεργασίες των στοιχείων ενός μονοδιάστατου πίνακα

Όπως έχουμε ήδη αναφέρει στις προηγούμενες ασκήσεις, όταν θέλουμε να εφαρμόσουμε μια επεξεργασία σε έναν πίνακα A με N θέσεις, τότε πρέπει αναγκαστικά να εφαρμόσουμε τη συγκεκριμένη επεξεργασία σε κάθε στοιχείο του πίνακα ξεχωριστά. Συγκεκριμένα πρέπει να επεξεργαστούμε αρχικά το πρώτο στοιχείο του πίνακα, στη συνέχεια το δεύτερο κ.λπ. Δηλαδή:

επεξεργασία του $A[1]$
επεξεργασία του $A[2]$
επεξεργασία του $A[3]$
.....
επεξεργασία του $A[N]$

Αν λοιπόν επιτρέψουμε σε μία μεταβλητή έστω i να πάρει διαδοχικά τις τιμές του πρώτου έως και του τελευταίου δείκτη του πίνακα, τότε μπορούμε να επεξεργαστούμε διαδοχικά κάθε στοιχείο $A[i]$ του πίνακα A με τη βοήθεια μιας επαναληπτικής δομής. Επειδή το πλήθος των στοιχείων που θέλουμε να επεξεργαστούμε είναι συγκεκριμένο, η δομή **για ... από ... μέχρι** είναι η προτιμότερη. Έτσι, αντί της παραπάνω διαδικασίας μπορούμε απλώς να γράψουμε το ακόλουθο τμήμα αλγόριθμου:

για i **από** 1 **μέχρι** N
 επεξεργασία του $A[i]$
τέλος_επανάληψης

Παραδείγματα στοιχειώδους επεξεργασίας πίνακα αποτελούν το διάβασμα και η εμφάνιση των στοιχείων του όπως θα δούμε στο θέμα 17.10. Αντί για τη μεταβλητή i που παίζει τον ρόλο μετρητή μπορούμε να χρησιμοποιήσουμε οποιοδήποτε άλλο όνομα μεταβλητής. Σε επόμενα κεφάλαια θα δούμε ότι η επιλογή κατάλληλου ονόματος για τον μετρητή διευκολύνει σημαντικά στην επίλυση των ασκήσεων.

17.10 Να γραφεί πρόγραμμα το οποίο θα διαβάζει έναν πίνακα A 100 θέσεων που θα περιέχει ακέραιους αριθμούς και απλώς θα τους εμφανίζει στην οθόνη.

Απάντηση

Το πρόγραμμα αυτό θα πρέπει αρχικά να διαβάζει τον πίνακα και, αφού ολοκληρωθεί η συγκεκριμένη επεξεργασία, να εμφανίζει τα στοιχεία του. Το διάβασμα του πίνακα θα πραγματοποιηθεί μέσω του διαβάσματος του κάθε στοιχείου του ξεχωριστά. Δηλαδή θα πρέπει να εκτελεστούν οι εντολές:

διάβασε A[1]

διάβασε A[2]

.....

διάβασε A[100]

Από τις παραπάνω εντολές βλέπουμε ότι το μόνο που αλλάζει είναι ο δείκτης των στοιχείων του πίνακα. Αν λοιπόν επιτρέψουμε σε μια μεταβλητή να πάρει διαδοχικά τις τιμές αυτές, τότε το σύνολο των παραπάνω εντολών μπορεί να εκτελεστεί με τη βοήθεια της επαναληπτικής δομής **για ... από ... μέχρι** ως εξής:

για κ από 1 μέχρι 100

διάβασε A[κ] ! Ζητάει μία τιμή από τον χρήστη και την καταχωρεί στη θέση
 ! του πίνακα που ορίζεται από το περιεχόμενο της μεταβλητής κ.

τέλος_επανάληψης

Αν ακολουθήσουμε την ίδια ακριβώς λογική και για την εμφάνιση των στοιχείων του πίνακα, έχουμε το ακόλουθο πρόγραμμα:

Πρόγραμμα Διάβασε_εμφάνισε_πίνακα

Μεταβλητές

ακέραιες: A[100], κ ! Δεν ξεχνάμε να δηλώσουμε και τον μετρητή.

Αρχή

 ! Διάβασμα των στοιχείων του πίνακα.

για κ από 1 μέχρι 100

διάβασε A[κ]

τέλος_επανάληψης

 ! Εμφάνιση των στοιχείων του πίνακα.

για κ από 1 μέχρι 100

γράψε A[κ]

τέλος_επανάληψης

Τέλος_Προγράμματος

Σημείωση

Για την επεξεργασία ενός πίνακα δεν είναι υποχρεωτικό να χρησιμοποιήσουμε την επαναληπτική δομή **για ... από ... μέχρι**, όμως ο συγκεκριμένος αριθμός επαναλήψεων την καθιστά πιο βολική για την απλότητα του αλγόριθμού μας. Ας δούμε πώς γίνεται το διάβασμα των στοιχείων στο παραπάνω πρόγραμμα χρησιμοποιώντας τη δομή **όσο ... επανάλαβε**.

```
κ ← 1
όσο κ <= 100 επανάλαβε
    διάβασε Α[κ]
    κ ← κ + 1
τέλος_επανάληψης
```

17.11 Να γίνει πρόγραμμα που θα διαβάσει 100 λέξεις και θα τις εμφανίζει με αντίθετη σειρά από αυτήν που διαβάστηκαν.

Απάντηση

Η εκφώνηση της άσκησης αυτής δεν μας ζητάει υποχρεωτικά να χρησιμοποιήσουμε τη δομή πίνακα. Για παράδειγμα, αν χρησιμοποιήσουμε τις γνώσεις που αποκτήσαμε από τις επαναληπτικές δομές, μπορούμε να γράψουμε το ακόλουθο τμήμα αλγόριθμου, το οποίο αναγκάζει τον χρήστη να δώσει 100 λέξεις:

```
για κ από 1 μέχρι 100
    διάβασε λέξη
τέλος_επανάληψης
```

Όταν όμως αυτό το τμήμα αλγόριθμου τερματίσει, στη μεταβλητή λέξη θα βρίσκεται μόνο η τελευταία λέξη που έδωσε ο χρήστης. Οι υπόλοιπες 99 χάθηκαν και δεν υπάρχει η δυνατότητα να τις εμφανίσουμε. Έτσι, λοιπόν, αντί να αποθηκεύσουμε τις λέξεις του χρήστη σε μεταβλητές, μπορούμε να τις αποθηκεύσουμε σε πίνακα 100 θέσεων, που σε κάθε θέση του θα καταχωρίζεται και η αντίστοιχη λέξη. Στη συνέχεια, για να εμφανιστούν οι λέξεις εισόδου ανάποδα από τη σειρά που διαβάστηκαν, αρκεί να εμφανιστεί πρώτα το τελευταίο στοιχείο του πίνακα, μετά το προτελευταίο κ.ο.κ. μέχρι στο τέλος να εμφανιστεί το πρώτο στοιχείο του που αντιστοιχεί στην πρώτη λέξη που δόθηκε από τον χρήστη. Αυτό μπορεί να επιτευχθεί εύκολα, αν με χρήση επαναληπτικής δομής υποχρεώσουμε τον μετρητή κ να πάρει διαδοχικά τις τιμές 100, 99, 98, ..., μέχρι 1 και για κάθε τιμή του να εμφανίζουμε το περιεχόμενο της αντίστοιχης θέσης του πίνακα. Το πρόγραμμα λοιπόν διαμορφώνεται ως εξής:

Πρόγραμμα Αντίθετη_Φορά

Μεταβλητές

χαρακτήρες: Λέξη[100]

ακέραιες: i

Αρχή

! Εισαγωγή των στοιχείων του πίνακα Λέξη.

για i από 1 μέχρι 100

διάβασε Λέξη[i]

τέλος_επανάληψης

! Εμφάνιση των στοιχείων με την αντίθετη φορά.
για i από 100 μέχρι 1 με_βήμα -1
 γράψε Λέξη[i]
 τέλος_επανάληψης
Τέλος_Προγράμματος

17.12 Να γίνει πρόγραμμα το οποίο θα διαβάξει έναν πίνακα A , 255 θέσεων, ο οποίος θα περιέχει ακέραιους αριθμούς και στη συνέχεια θα εμφανίζει τους θετικούς αριθμούς που δόθηκαν.

Απάντηση

Ο τρόπος εμφάνισης πίνακα που ζητείται εδώ διαφέρει από τη συνηθισμένη εμφάνιση πίνακα (δες, για παράδειγμα, θέμα 17.10) στο ότι, αν και πρέπει να διατρέξουμε όλα τα στοιχεία του πίνακα, δεν θα εμφανίζουμε το καθένα από αυτά αλλά μόνον αυτά που είναι θετικά. Δηλαδή η εντολή **γράψε $A[k]$** θα περιβάλλεται από μια δομή επιλογής **αν ... τότε**, η οποία θα ελέγχει αν το συγκεκριμένο στοιχείο είναι θετικό οπότε και θα το εμφανίζει, διαφορετικά θα το αγνοεί. Συγκεκριμένα το πρόγραμμα έχει ως εξής:

Πρόγραμμα Εμφάνισε_θετικούς

Μεταβλητές

 ακέραιες: $A[255]$, k

Αρχή

 ! Διάβασμα των στοιχείων του πίνακα.

 για k από 1 μέχρι 255

 διάβασε $A[k]$

 τέλος_επανάληψης

 ! Εμφάνιση των θετικών στοιχείων του πίνακα.

 για k από 1 μέχρι 255

 αν $A[k] > 0$ τότε

 γράψε $A[k]$

 τέλος

 τέλος_επανάληψης

Τέλος_Προγράμματος

17.13 Να γίνει αλγόριθμος που θα διαβάξει έναν πίνακα 100 θέσεων ο οποίος θα περιέχει ακέραιους αριθμούς και θα εμφανίζει τους άρτιους αριθμούς που καταχωρίστηκαν σε περιττές θέσεις.

Απάντηση

Αν B το όνομα του πίνακα και i μία θέση του, για να εμφανίσουμε το περιεχόμενο της θέσης αυτής, θα πρέπει:

- ▶ η θέση i να είναι περιττός αριθμός, δηλαδή $i \bmod 2 = 1$

► και συγχρόνως το περιεχόμενο αυτής της θέσης να είναι άρτιος αριθμός, δηλαδή $B[i] \bmod 2 = 0$

Ο αλγόριθμος, λοιπόν, έχει ως εξής:

Αλγόριθμος Άρτιοι

για i από 1 μέχρι 100

διάβασε $B[i]$

τέλος_επανάληψης

! Για (κάθε θέση) i από (την) 1η μέχρι (και την) 100ή.

για i από 1 μέχρι 100

αν $i \bmod 2 = 1$ και $B[i] \bmod 2 = 0$ τότε

εμφάνισε $B[i]$

τέλος_αν

τέλος_επανάληψης

Τέλος Άρτιοι

Σημείωση

Για να εμφανίσουμε τα στοιχεία που βρίσκονται σε περιττές θέσεις και είναι άρτια, μπορούμε ισοδύναμα να γράψουμε και το ακόλουθο τμήμα:

για i από 1 μέχρι 99 με_βήμα 2

! Ο μετρητής παίρνει μόνο τις περιττές τιμές.

αν $B[i] \bmod 2 = 0$ τότε

εμφάνισε $B[i]$

τέλος_αν

τέλος_επανάληψης

17.14 Να γίνει αλγόριθμος που για δεδομένο έναν πίνακα N θέσεων με ακέραιες τιμές θα εμφανίζει τις θέσεις του που περιέχουν άρτιες τιμές.

Απάντηση

Ερώτηση

Τι σημαίνει για έναν αλγόριθμο η φράση «με δεδομένο έναν πίνακα N θέσεων»; Μήπως πρέπει ο χρήστης να μας δώσει το μέγεθος N του πίνακα;

Απάντηση

Όπως είπαμε, το μέγεθος του πίνακα είναι πάντα σταθερό και καθορίζεται στο τμήμα δηλώσεων του προγράμματος. Έτσι, όταν η εκφώνηση μιας άσκησης αναφέρει με δεδομένο έναν πίνακα N θέσεων, αυτό σε καμία περίπτωση δεν σημαίνει ότι το μέγεθος του πίνακα είναι μεταβλητό και εξαρτάται από τον χρήστη. Πρέπει να θυμηθούμε ότι ο αλγόριθμος σε μορφή ψευδογλώσσας απευθύνεται σε έναν προγραμματιστή που θα πάρει τον αλγόριθμο και εύκολα θα τον υλοποιήσει στη γλώσσα προγραμματισμού που αυτός επιθυμεί. Όταν, λοιπόν, στην εκφώνηση της άσκησης

λέει με δεδομένο έναν πίνακα N θέσεων, αυτό σημαίνει ότι δεν χρειάζεται να περιγράψουμε τη διαδικασία εισαγωγής τιμών στον πίνακα, αφού αυτή θεωρείται γνωστή από τον προγραμματιστή, αλλά μόνο την επεξεργασία που πρέπει να υποστούν τα στοιχεία του. Αρκεί, λοιπόν, στον αλγόριθμό μας να γράψουμε **δεδομένα // A, N //** και από εκεί και πέρα να αναπτύξουμε τις διαδικασίες που ζητά η εκφώνηση θεωρώντας ότι ο πίνακας και η μεταβλητή N έχουν πάρει τιμές.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Θέσεις_με_άρτιους
δεδομένα // A, N //
για  $i$  από 1 μέχρι N
    αν  $A[i] \bmod 2 = 0$  τότε ! Αν το περιεχόμενο του πίνακα A στη θέση  $i$  είναι
                                ! άρτιος αριθμός.
        εμφάνισε  $i$  ! Προσοχή! Θέλουμε να εμφανιστεί η θέση και όχι
        τέλος_αν ! το  $A[i]$ .
    τέλος_επανάληψης
Τέλος Θέσεις_με_άρτιους

```

Δεν θα ήταν λάθος αν ένας μαθητής έγραφε αντί για **δεδομένα // A, N //** το παρακάτω τμήμα αλγόριθμου:

```

δεδομένα // N //
για  $i$  από 1 μέχρι N
    διάβασε  $A[i]$ 
τέλος_επανάληψης

```

Όμως **προσοχή**, θα ήταν λάθος αντί για **δεδομένα // N //** να βάζαμε **διάβασε N**, γιατί τότε ζητάμε από τον χρήστη να μας δώσει το μέγεθος του πίνακα.

17.15 Να γίνει αλγόριθμος που για δεδομένο έναν πίνακα N θέσεων με πραγματικούς αριθμούς θα εμφανίζει το άθροισμα των στοιχείων του.

Απάντηση

Για το άθροισμα των στοιχείων ενός πίνακα αρκεί να δουλέψουμε με τρόπο ανάλογο με το άθροισμα αριθμών στην ενότητα των επαναληπτικών δομών. Συγκεκριμένα μια μεταβλητή έστω Σ που αρχικά έχει την τιμή 0 πρώτα θα αυξηθεί κατά το περιεχόμενο της πρώτης θέσης του πίνακα, στη συνέχεια κατά το περιεχόμενο της δεύτερης κ.ο.κ. Δηλαδή θα πρέπει να εκτελεστούν οι εντολές:

```

 $\Sigma \leftarrow 0$ 
 $\Sigma \leftarrow \Sigma + A[1]$ 
 $\Sigma \leftarrow \Sigma + A[2]$ 
.....
 $\Sigma \leftarrow \Sigma + A[N]$ 

```

Όπως είδαμε, αυτή η σειρά εντολών μπορεί να εκτελεστεί με την επαναληπτική δομή **για ... από ... μέχρι**. Ο αλγόριθμος, λοιπόν, είναι ο ακόλουθος:

Αλγόριθμος Άθροισμα

δεδομένα // A, N //

$\Sigma \leftarrow 0$

! Για (κάθε θέση) κ από (την) 1η μέχρι (και τη) Νή.

για κ **από** 1 **μέχρι** N

$\Sigma \leftarrow \Sigma + A[\kappa]$

τέλος_επανάληψης

εμφάνισε Σ

Τέλος Άθροισμα

17.16 Να γίνει αλγόριθμος ο οποίος θα διαβάζει έναν πίνακα A 100 θέσεων με ακέραιους αριθμούς και θα δημιουργεί έναν πίνακα B 50 θέσεων, στον οποίο θα καταχωριστούν τα στοιχεία του πίνακα A που βρίσκονται σε άρτιες θέσεις.

Απάντηση

Ο πίνακας B είναι 50 θέσεων, τόσων δηλαδή θέσεων όσο και το πλήθος των στοιχείων του πίνακα A που βρίσκονται σε άρτιες θέσεις. Κατά την εκτέλεση του αλγόριθμου πρέπει στο B[1] να καταχωριστεί το A[2], στο B[2] το A[4], στο B[3] το A[6] και γενικότερα στο B[κ] το A[2 * κ]. Ο αλγόριθμος, λοιπόν, είναι ο ακόλουθος:

Αλγόριθμος Άρτιες_θέσεις

! Εισαγωγή των στοιχείων του πίνακα A.

για κ **από** 1 **μέχρι** 100

διάβασε A[κ]

τέλος_επανάληψης

! Δημιουργία του πίνακα B.

για κ **από** 1 **μέχρι** 50

$B[\kappa] \leftarrow A[2 * \kappa]$

τέλος_επανάληψης

αποτελέσματα // B //

Τέλος Άρτιες_θέσεις

Σημείωση

Η εντολή **αποτελέσματα** // B // σε αντιστοιχία με τη **δεδομένα** // A // σημαίνει ότι ο αλγόριθμος αυτός δίνει ως αποτέλεσμα τον πίνακα B. Ο προγραμματιστής, δηλαδή, που θα διαβάσει τον αλγόριθμό μας και θα τον μετατρέψει σε μια γλώσσα προγραμματισμού θα πάρει τον πίνακα B και θα τον χειριστεί όπως αυτός επιθυμεί. Δεν είναι αναγκαστικό ότι θα τον εμφανίσει στην οθόνη.

17.17 Να γίνει πρόγραμμα που θα διαβάζει 100 αριθμούς και θα τους καταχωρίζει σε πίνακα A, 100 θέσεων. Να δημιουργεί και να εμφανίζει έναν πίνακα B, 99 θέσεων, που κάθε στοιχείο του B[i] ισούται με:

$$\frac{A[i] + A[i + 1]}{2}$$

Απάντηση

Ας μελετήσουμε πρώτα την πρόταση της εκφώνησης: **Να γίνει πρόγραμμα που θα διαβάζει 100 αριθμούς και θα τους καταχωρίζει σε πίνακα A, 100 θέσεων.**

Πολλοί μαθητές νομίζουν ότι αυτή η πρόταση εννοεί ότι πρέπει πρώτα να διαβάσουμε ένα στοιχείο το οποίο θα αποθηκευτεί σε μια μεταβλητή και στη συνέχεια το περιεχόμενο της μεταβλητής να το καταχωρίσουμε στον πίνακα. Δηλαδή να εκτελέσουμε το ακόλουθο τμήμα αλγόριθμου:

για i από 1 μέχρι 100

διάβασε x

A[i] ← x

τέλος_επανάληψης

Αν και αλγοριθμικά η παραπάνω αντιμετώπιση δεν είναι λάθος, τα πράγματα είναι πολύ πιο απλά. Αρκεί να θυμηθούμε το πώς μεταφράζεται η εντολή **διάβασε A[k]**: Ζητά μία τιμή από τον χρήστη και καταχωρίσέ τη στη θέση k του πίνακα A. Δηλαδή διάβασε μια τιμή και καταχώρισέ τη στη θέση k του πίνακα, πράγμα που είναι ακριβώς αυτό που ζητάει και η εκφώνηση. Έτσι, μπορούμε να γράψουμε πιο απλά:

για i από 1 μέχρι 100

διάβασε A[i]

τέλος_επανάληψης

Το ζητούμενο πρόγραμμα είναι το ακόλουθο:

Πρόγραμμα Άσκηση17_17

Μεταβλητές

πραγματικές: A[100], B[99]

ακέραιες: i ! Δεν ξεχνάμε να δηλώσουμε και τον μετρητή.

Αρχή

! Διάβασμα του πίνακα A.

για i από 1 μέχρι 100

διάβασε A[i]

τέλος_επανάληψης

! Δημιουργία του πίνακα B.

για i από 1 μέχρι 99

```
B[i] ← (A[i] + A[i + 1]) / 2
τέλος_επανάληψης
! Εμφάνιση του πίνακα B.
για i από 1 μέχρι 99
    γράψε B[i]
τέλος_επανάληψης
Τέλος_Προγράμματος
```

Ερώτηση

Θα μπορούσε η δημιουργία και η εμφάνιση του πίνακα B να γίνουν με τη βοήθεια μιας επανάληψης όπως:

```
για i από 1 μέχρι 99
    B[i] ← (A[i] + A[i + 1]) / 2
    γράψε B[i]
τέλος_επανάληψης
```

Απάντηση

Βεβαίως, αλλά προτείνουμε για απλότητα κάθε επεξεργασία του πίνακα να γίνεται σε διαφορετική επαναληπτική δομή.

17.18 Να γίνει αλγόριθμος ο οποίος θα διαβάζει 100 θετικούς αριθμούς και θα τους καταχωρίζει σε πίνακα 100 θέσεων, κάνοντας παράλληλα έλεγχο εγκυρότητας, ώστε οι αριθμοί εισόδου να είναι θετικοί. Στη συνέχεια, να εμφανίζει τον μέσο όρο όλων των στοιχείων του πίνακα.

Απάντηση

Ο αλγόριθμος αυτός θα πρέπει να επιτρέπει την εισαγωγή 100 τιμών από τον χρήστη κάνοντας παράλληλα έλεγχο εγκυρότητας, ώστε οι αριθμοί αυτοί να είναι θετικοί. Επίσης θα πρέπει αυτούς τους αριθμούς να τους καταχωρίζει σε πίνακα 100 θέσεων (μία θέση για κάθε έγκυρο αριθμό). Ένα τμήμα αλγόριθμου που μας εξασφαλίζει τα παραπάνω είναι το ακόλουθο:

```
για κ από 1 μέχρι 100
    διάβασε x
    όσο x < 0 επανάλαβε
        εμφάνισε “Απαγορεύονται αρνητικοί. Ξαναδώσε αριθμό.”
        διάβασε x
    τέλος_επανάληψης
    A[κ] ← x
τέλος_επανάληψης
```


Με αυτόν τον τρόπο η εισαγωγή και ο έλεγχος μιας τιμής γίνεται με τη βοήθεια μιας μεταβλητής x . Αφού ο αριθμός εισόδου ελέγχθηκε ότι είναι έγκυρος, αποθηκεύεται στον πίνακα στη θέση που ορίζει ο μετρητής k . Όπως είδαμε στο θέμα 17.17, η εισαγωγή στοιχείων σε έναν πίνακα δεν απαιτεί κάποια βοηθητική μεταβλητή x αλλά πραγματοποιείται απευθείας με εντολή της μορφής **διάβασε** $A[k]$. Με ανάλογο τρόπο μπορεί να υλοποιηθεί και η εισαγωγή τιμής με έλεγχο εγκυρότητας πιο απλά, όπως φαίνεται παρακάτω:

για k **από** 1 **μέχρι** 100

διάβασε $A[k]$

όσο $A[k] < 0$ **επανάλαβε**

εμφάνισε “Απαγορεύονται αρνητικοί. Ξαναδώσε αριθμό.”

διάβασε $A[k]$

τέλος_επανάληψης

τέλος_επανάληψης

Δηλαδή αποθηκεύεται ένας αριθμός στη θέση k του πίνακα και ελέγχεται για το αν είναι έγκυρος ή όχι. Αν δεν είναι, τότε στην ίδια θέση καταχωρίζεται ένας νέος αριθμός από τον χρήστη, ο οποίος ελέγχεται ξανά κ.ο.κ.

Όσον αφορά τον μέσο όρο, αρκεί να υπολογίσουμε το άθροισμα όλων των στοιχείων του πίνακα (δες θέμα 17.15) και στη συνέχεια να διαιρέσουμε με το πλήθος τους, που στο συγκεκριμένο θέμα είναι γνωστό και ίσο με 100. Ο αλγόριθμος, λοιπόν, του προβλήματος είναι ο ακόλουθος:

Αλγόριθμος Άσκηση

! Εισαγωγή και έλεγχος 100 αριθμών.

για k **από** 1 **μέχρι** 100

διάβασε $A[k]$

όσο $A[k] < 0$ **επανάλαβε**

εμφάνισε “Απαγορεύονται αρνητικοί. Ξαναδώσε αριθμό.”

διάβασε $A[k]$

τέλος_επανάληψης

τέλος_επανάληψης

! Υπολογισμός του μέσου όρου.

$\Sigma \leftarrow 0$

για k **από** 1 **μέχρι** 100

$\Sigma \leftarrow \Sigma + A[k]$

τέλος_επανάληψης

$MO \leftarrow \Sigma / 100$

εμφάνισε MO

Τέλος Άσκηση

17.19 Να γίνει αλγόριθμος ο οποίος θα διαβάζει 100 άρτιους αριθμούς και θα τους καταχωρίζει σε πίνακα 100 θέσεων, κάνοντας παράλληλα έλεγχο εγκυρότητας. Στη συνέχεια, θα εμφανίζει τον μέσο όρο όλων των στοιχείων του πίνακα που είναι μεγαλύτερα του 500.

Απάντηση

Όσον αφορά τον έλεγχο εγκυρότητας, δουλεύουμε με τρόπο ανάλογο με αυτόν του προηγούμενου θέματος. Για τον μέσο όρο, όπως και σε αντίστοιχα θέματα των επαναληπτικών δομών, θα πρέπει να υπολογίζουμε εκτός του αθροίσματος και το πλήθος των αριθμών που προστέθηκαν στο άθροισμα, δηλαδή το πλήθος αυτών που είναι μεγαλύτεροι του 500.

Τέλος, θα ακολουθήσει η διαίρεση του αθροίσματος με το πλήθος των αριθμών, εξασφαλίζοντας πρώτα ότι το πλήθος αυτών δεν είναι 0 (δες θέμα 12.6). Ο αλγόριθμος, λοιπόν, είναι ο ακόλουθος:

Αλγόριθμος MO500

! Εισαγωγή και έλεγχος 100 αριθμών.

για κ από 1 μέχρι 100

 διάβασε A[κ]

 όσο A[κ] mod 2 <> 0 επανάλαβε

 εμφάνισε “Λανθασμένη είσοδος. Δώσε άρτιο αριθμό.”

 διάβασε A[κ]

 τέλος_επανάληψης

τέλος_επανάληψης

! Υπολογισμός του μέσου όρου των μεγαλύτερων του 500.

Σ ← 0

πλήθος ← 0

για κ από 1 μέχρι 100

 αν A[κ] > 500 τότε

 Σ ← Σ + A[κ]

 πλήθος ← πλήθος + 1

 τέλος_αν

τέλος_επανάληψης

! Προσοχή. Δεν βιαζόμαστε να γράψουμε MO ← Σ / πλήθος.

αν πλήθος > 0 τότε

 MO ← Σ / πλήθος

 εμφάνισε MO

αλλιώς

 εμφάνισε “Δεν υπάρχουν αριθμοί μεγαλύτεροι του 500.”

τέλος_αν

Τέλος MO500

17.20 Να γίνει πρόγραμμα το οποίο θα ζητά από τον χρήστη το πλήθος των αριθμών που ο ίδιος θα δώσει και στη συνέχεια θα διαβάζει τους αριθμούς αυτούς καταχωρίζοντάς τους σε πίνακα. Αν το πλήθος των αριθμών εισόδου δεν μπορεί να υπερβαίνει το 1.000, ο αλγόριθμος θα πρέπει να εμφανίζει το πλήθος των θετικών αριθμών που καταχωρίστηκαν στον πίνακα και στη συνέχεια θα τερματίζει. Να σημειωθεί ότι αρχικά θα πρέπει να γίνεται έλεγχος εγκυρότητας, ώστε το πλήθος των αριθμών να είναι μεγαλύτερο του μηδενός και μικρότερο ή ίσο του 1.000.

Απάντηση

Το πρόγραμμα αυτό θα πρέπει να δέχεται από τον χρήστη έναν αριθμό ο οποίος θα εκφράζει το πλήθος των αριθμών που θα δώσει ο χρήστης στη συνέχεια. Αυτοί οι αριθμοί θα καταχωριστούν σε πίνακα. Ένα πρόβλημα που παρουσιάζεται είναι ότι το μέγεθος ενός πίνακα δεν μπορεί να είναι μεταβλητό. Με άλλα λόγια, δεν έχουμε την πολυτέλεια να περιμένουμε να δώσει ο χρήστης το πλήθος των τιμών εισόδου και στη συνέχεια να δημιουργήσουμε έναν πίνακα που θα έχει τον ακριβή αριθμό θέσεων. Το μέγεθος του πίνακα πρέπει υποχρεωτικά να καθοριστεί στο τμήμα δηλώσεων του προγράμματος. Παρ' όλα αυτά, μπορούμε εξ αρχής να χρησιμοποιήσουμε έναν πίνακα 1.000 στοιχείων, μιας και το πλήθος των τιμών εισόδου δεν μπορεί να ξεπεράσει το 1.000, αφήνοντας πιθανώς κάποιες θέσεις αχρησιμοποίητες. Για παράδειγμα, αν ο χρήστης θέλει να δώσει 50 αριθμούς, τότε θα καταληφθούν μόνο οι 50 πρώτες θέσεις του πίνακα, αφήνοντας τις υπόλοιπες 950 με τις απροσδιόριστες τιμές που αρχικά περιείχε ολόκληρος ο πίνακας. Από εκεί και πέρα η επεξεργασία του πίνακα θα αφορά μόνο αυτές τις πρώτες 50 θέσεις.

Πρόγραμμα Θετικοί

Μεταβλητές

ακέραιοι: A[1000], κ, πλ_αριθμών, πλ_θετικών

Αρχή

! Εισαγωγή του πλήθους των αριθμών που θα δώσει στη συνέχεια ο χρήστης.

αρχή_επανάληψης

διάβασε πλ_αριθμών

μέχρις_ότου πλ_αριθμών > 0 και πλ_αριθμών <= 1000

! Εισαγωγή τόσων αριθμών όσους ορίζει το περιεχόμενο της πλ_αριθμών.

για κ από 1 μέχρι πλ_αριθμών

διάβασε A[κ]

τέλος_επανάληψης

! Υπολογισμός του πλήθους των θετικών αριθμών εισόδου.

πλ_θετικών ← 0

για κ από 1 μέχρι πλ_αριθμών

αν A[κ] > 0 τότε

πλ_θετικών ← πλ_θετικών + 1

τέλος_αν
τέλος_επανάληψης
γράψε πλ_θετικών
Τέλος_Προγράμματος

17.21 Να γίνει αλγόριθμος ο οποίος θα δημιουργεί τον ακόλουθο πίνακα 50 θέσεων.

	1	2	3	...	50
A	5	10	15	...	250

Απάντηση

Στον παραπάνω πίνακα βλέπουμε ότι στο $A[1]$ πρέπει να καταχωριστεί το 5, στο $A[2]$ το 10, στο $A[3]$ το 15 και γενικότερα στο $A[k]$ το $5 * k$. Ο αλγόριθμος, λοιπόν, είναι ο ακόλουθος:

Αλγόριθμος Άσκηση
για κ από 1 μέχρι 50
 $A[k] \leftarrow 5 * k$
 τέλος_επανάληψης
 αποτελέσματα // A //
Τέλος Άσκηση

17.22 Να γίνει αλγόριθμος ο οποίος θα δημιουργεί τον ακόλουθο πίνακα 100 θέσεων.

	1	2	3	...	100
A	2	5	11	...	$2 * A[99] + 1$

Σημείωση: Ο τύπος στην 100ή θέση μάς δείχνει τον τρόπο που συνδέεται κάθε στοιχείο του πίνακα με το προηγούμενό του, για παράδειγμα το $A[3]$ περιέχει το αποτέλεσμα της πράξης $2 * A[2] + 1$.

Απάντηση

Για την υλοποίηση του συγκεκριμένου αλγόριθμου θα πρέπει να προσέξουμε ότι το κάθε στοιχείο $A[k]$ υπολογίζεται με βάση το προηγούμενό του σύμφωνα με την εντολή $A[k] \leftarrow 2 * A[k - 1] + 1$. Αυτή η εντολή πρέπει να εκτελεστεί με τη βοήθεια επαναληπτικής δομής για κάθε θέση k του πίνακα, εκτός της πρώτης, αφού αυτή δεν έχει προηγούμενο στοιχείο. Η πρώτη θέση θα πρέπει να πάρει την τιμή 2 πριν από την επαναληπτική διαδικασία. Ο αλγόριθμος, λοιπόν, είναι ο ακόλουθος:

Αλγόριθμος Δημιουργία_Πίνακα

$A[1] \leftarrow 2$! Για το στοιχείο ενεργούμε διαφορετικά από τα υπόλοιπα.
για κ από 2 μέχρι 100
 $A[k] \leftarrow 2 * A[k - 1] + 1$

Προσοχή

Όταν σε κάποια άσκηση κάθε στοιχείο ενός πίνακα παίρνει τιμές με βάση τα στοιχεία που βρίσκονται σε προηγούμενες ή σε επόμενες θέσεις, θα πρέπει να δίνουμε ιδιαίτερη προσοχή στα ακριανά στοιχεία του πίνακα, δηλαδή στο πρώτο και το τελευταίο, γιατί αυτά, αφού δεν έχουν προηγούμενο και επόμενο στοιχείο αντίστοιχα, πιθανόν να θέλουν διαφορετική αντιμετώπιση από τα υπόλοιπα.

Ερωτήσεις και προβλήματα για απάντηση

17.23 Ερωτήσεις ανάπτυξης

1. Να δώσετε τον ορισμό του πίνακα.
2. Να δώσετε τον ορισμό του μονοδιάστατου πίνακα.
3. Ποια είναι τα χαρακτηριστικά των πινάκων και πού αυτά καθορίζονται;
4. Πώς δηλώνουμε έναν μονοδιάστατο πίνακα σε ένα πρόγραμμα;

Πριν λύσεις τις ασκήσεις 17.24-17.32 καλό είναι να διαβάσεις πρώτα την παράγραφο «Εισαγωγικά στοιχεία πινάκων-μονοδιάστατων πινάκων».

17.24 Να χαρακτηρίσετε ως σωστή (Σ) ή λάθος (Λ) καθεμία από τις παρακάτω προτάσεις.

1. Το όνομα ενός πίνακα ακολουθεί τους κανόνες ονοματολογίας που ακολουθεί και το όνομα μιας μεταβλητής. Σ Λ
2. Το μέγεθος ενός πίνακα καθορίζεται από τον χρήστη του προγράμματος. Σ Λ
3. Γράφοντας στο τμήμα δηλώσεων **ακέραιες**: $A[100]$, σημαίνει ότι ο πίνακας A έχει 100 θέσεις και σε καθεμία μπορούν να αποθηκευτούν ακέραιες τιμές. Σ Λ
4. Σε ένα πρόγραμμα δύο πίνακες δεν μπορούν να έχουν το ίδιο μέγεθος. Σ Λ
5. Σε ένα πρόγραμμα δύο πίνακες δεν μπορούν να έχουν το ίδιο όνομα. Σ Λ
6. Γράφοντας $B[5] \leftarrow 100$, αποθηκεύουμε στη θέση 5 του πίνακα την τιμή 100. Σ Λ
7. Ως δείκτη ενός πίνακα δεν μπορεί να έχουμε μεταβλητή. Σ Λ

8. Ως δείκτη ενός πίνακα μπορεί να έχουμε μαθηματική έκφραση. Σ Λ
9. Γράφοντας στο τμήμα δηλώσεων **πραγματικές**: $A[100]$, σημαίνει ότι ο δείκτης αναφοράς ενός στοιχείου του πίνακα μπορεί να είναι πραγματικός αριθμός από 1 μέχρι και 100. Σ Λ
10. Η αρχική τιμή για ένα στοιχείο του πίνακα πρέπει να θεωρείται το μηδέν. Σ Λ

17.25 Για τον διπλανό μονοδιάστατο πίνακα A να σημειώσετε τι εμφανίζουν τα παρακάτω τμήματα αλγόριθμων:

A	4	1	3	9	-1
---	---	---	---	---	----

- i. εμφάνισε $A[3]$ ii. εμφάνισε $A[2 * 2]$
- iii. $i \leftarrow 1$ iv. $i \leftarrow 1$
 $j \leftarrow 2$ **εμφάνισε** $A[A[i]]$
 εμφάνισε $A[i + j]$

17.26 Να γράψετε το τμήμα δήλωσης:

- i. για έναν πίνακα ο οποίος θα περιέχει 100 πραγματικούς αριθμούς,
 ii. για έναν πίνακα ο οποίος θα περιέχει το όνομα 100 μαθητών,
 iii. για έναν πίνακα ο οποίος θα περιέχει το ύψος 20 κορυφών της Ελλάδας,
 iv. για έναν πίνακα ο οποίος θα περιέχει το φύλο 300 μαθητών ενός σχολείου,
 v. για δύο πίνακες που ο πρώτος θα περιέχει 100 πραγματικούς αριθμούς και ο δεύτερος 200 πραγματικούς αριθμούς,
 vi. για έναν πίνακα ο οποίος θα περιέχει το πλήθος παιδιών καθεμιάς από 200 οικογένειες.

17.27 Να συμπληρώσετε τα στοιχεία του διπλανού πίνακα A μετά την εκτέλεση των παρακάτω εντολών:

A					
---	--	--	--	--	--

- $A[1] \leftarrow 2$
 $A[3] \leftarrow 5$
 $A[A[3]] \leftarrow A[3] \bmod 2$
 $A[4] \leftarrow A[1] \text{ div } 4$
 $A[A[4] + 2] \leftarrow A[4] * 3$

17.28 Να συμπληρώσετε τα κενά του επόμενου τμήματος αλγόριθμου, ώστε να δημιουργεί τον διπλανό πίνακα B.

B	23	11	25	0	-6
---	----	----	----	---	----

```
B[5] ← ...
x ← ...
B[x] ← 11
B[x - ...] ← ...
B[x + 1] ← B[x] + ...
B[x + ...] ← B[x - ...] * 0
```

17.29 Έστω το παρακάτω πρόγραμμα:

Πρόγραμμα Άσκηση

Μεταβλητές

ακέραιες: A[5]

Αρχή

διάβασε A[2]

A[1] ← A[2] * 10

διάβασε A[3]

A[4] ← A[3] mod 5

διάβασε A[2]

A[1] ← A[2] - 3

αν A[1] mod 2 = 0 τότε

A[5] ← A[1]

αλλιώς

A[5] ← A[2]

τέλος_αν

Τέλος_Προγράμματος

Να συμπληρώσετε τα στοιχεία του διπλανού πίνακα A, αν ο χρήστης δώσει με τη σειρά τις τιμές 5, 8 και 11.

A

--	--	--	--	--

17.30 Έστω το παρακάτω πρόγραμμα:

Πρόγραμμα Άσκηση

Μεταβλητές

χαρακτήρες: B[5]

Αρχή

διάβασε B[1]

B[2] ← “Νίκος”

B[3] ← B[1]

B[4] ← “B[2]”

αν B[3] >= “Γ” και B[3] < “Δ” τότε

B[5] ← “Μαρία”

αλλιώς

B[5] ← “Εύα”

τέλος_αν

Τέλος_Προγράμματος

Να συμπληρώσετε τα στοιχεία του διπλανού πίνακα Β, αν:

B

--	--	--	--	--

- i. ο χρήστης δώσει την τιμή “Μιχάλης”,
- ii. ο χρήστης δώσει την τιμή “Γιώργος”.

17.31 Έστω το παρακάτω πρόγραμμα:

Πρόγραμμα Άσκηση

Μεταβλητές

ακέραιες: Γ[5], x

Αρχή

x ← 5

Γ[x] ← x

Γ[x - 1] ← 4

Γ[1] ← Γ[4] + Γ[5]

Γ[2] ← x - 6

x ← x + Γ[1]

Γ[3] ← x

Γ[2] ← Γ[2] + Γ[3]

Τέλος_Προγράμματος

Να γράψετε για κάθε βήμα του αλγόριθμου τι τιμές έχει κάθε στοιχείο του πίνακα καθώς και η μεταβλητή x.

17.32 Δίνεται το παρακάτω τμήμα αλγόριθμου με αριθμημένες εντολές για εύκολη αναφορά σε αυτές. Κάθε εντολή περιέχει ένα ή δύο κενά (σημειωμένα με ...), που το καθένα αντιστοιχεί σε μία σταθερά ή μία μεταβλητή ή έναν τελεστή. Επίσης δίνεται πίνακας όπου κάθε γραμμή αντιστοιχεί στη διπλανή εντολή του τμήματος αλγόριθμου και κάθε στήλη σε μία θέση μνήμης (μεταβλητή). Η κάθε γραμμή του πίνακα παρουσιάζει το αποτέλεσμα που έχει η εκτέλεση της αντίστοιχης εντολής στη μνήμη: συγκεκριμένα, δείχνει την τιμή της μεταβλητής την οποία επηρεάζει η εντολή.

	Εντολές	Μνήμη								
		A	B	Γ	Δ	E	Z	X[1]	X[2]	X[3]
1.	A ← ...	4								
2.	Δ ← A + ...				7					
3.	αν A ... Δ τότε									

	Εντολές	Μνήμη								
		A	B	Γ	Δ	E	Z	X[1]	X[2]	X[3]
	$\Gamma \leftarrow A$ αλλιώς $\Gamma \leftarrow \Delta$ τέλος_αν			7						
4.	$B \leftarrow \dots - 1$		3							
5.	$E \leftarrow \dots - \dots$					-1				
6.	$\dots \leftarrow \Delta + \dots$				6					
7.	$\Gamma \leftarrow \Gamma \dots E$			8						
8.	$Z \leftarrow \dots - 1$						2			
9.	$X[\dots] \leftarrow \Gamma$								8	
10.	$X[Z \dots 1] \leftarrow \Delta$							6		
11.	$X[Z \dots 1] \leftarrow X[Z] \dots 1$									7

Να γράψετε τον αριθμό της καθεμιάς εντολής και δίπλα να σημειώσετε τη σταθερά, τη μεταβλητή ή τον τελεστή που πρέπει να αντικαταστήσει το κάθε κενό της εντολής, ώστε να έχει το αποτέλεσμα που δίνεται στον πίνακα, ως εξής:

- i. Για τις εντολές 1 και 2 να σημειώσετε σταθερές τιμές.
- ii. Για τις εντολές 3, 7, 10 και 11 να σημειώσετε τελεστές και για τις υπόλοιπες να σημειώσετε μεταβλητές. (Εξετάσεις 2007)

Πριν λύσεις τις ασκήσεις 17.33-17.62 καλό είναι να διαβάσεις πρώτα την παράγραφο «Βασικές επεξεργασίες των στοιχείων ενός μονοδιάστατου πίνακα».

17.33 Να χαρακτηρίσετε ως σωστή (Σ) ή λάθος (Λ) καθεμία από τις παρακάτω προτάσεις.

1. Γράφοντας **δεδομένα** // A, N //, όπου A ένας πίνακας μεγέθους N, δηλώνουμε ότι ο πίνακας A δεν έχει σταθερό μέγεθος. Σ Λ
2. Η επεξεργασία ενός πίνακα γίνεται για καθένα στοιχείο του ξεχωριστά. Σ Λ
3. Για να διαβάσουμε έναν πίνακα A, 10 θέσεων, γράφουμε **διάβασε** A. Σ Λ
4. Όλα τα στοιχεία ενός πίνακα πρέπει να είναι του ίδιου τύπου. Σ Λ
5. Η δήλωση **ακέραιες**: A[i] έχει νόημα. Σ Λ
6. Αν ως δείκτη για τη θέση ενός πίνακα χρησιμοποιήσουμε μεταβλητή, τότε θα πρέπει να είναι ακέραιος αριθμός. Σ Λ

7. Ο μετρητής που χρησιμοποιούμε στη **για ... από ... μέχρι**, για να διαβάσουμε τα στοιχεία ενός πίνακα πρέπει να είναι πάντα είτε ο **κ** ή ο **i**. Σ Λ
8. Για να διαβάσουμε τα στοιχεία ενός πίνακα, η μόνη δομή επανάληψης την οποία μπορούμε να χρησιμοποιήσουμε είναι η **για ... από ... μέχρι**. Σ Λ
9. Όλα τα στοιχεία ενός πίνακα πρέπει να είναι διαφορετικά μεταξύ τους. Σ Λ
10. Αν γράψω την εντολή $A[0] \leftarrow 8$, δεν ικανοποιείται το κριτήριο της καθοριστικότητας. Σ Λ

17.34 Δίνεται μονοδιάστατος πίνακας B, 8 θέσεων, ο οποίος στις θέσεις 1 έως 8 περιέχει αντίστοιχα τους αριθμούς 2, 0, 0, 4, 11, 0, 1, 3, καθώς και το παρακάτω τμήμα αλγόριθμου:

```

για j από 1 μέχρι 8
  αν B[j] = 0 τότε
    B[j] ← B[j] + 1
  αλλιώς
    B[j] ← B[j] - 1
τέλος_αν
τέλος_επανάληψης

```

Να γράψετε τι τιμή θα έχει κάθε στοιχείο του πίνακα μετά την εκτέλεση του παραπάνω τμήματος.

17.35 Να συμπληρώσετε τα στοιχεία του πίνακα A:

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]

μετά την εκτέλεση του ακόλουθου τμήματος αλγόριθμου, αν ο χρήστης δώσει τις τιμές 5, 10, 0 και 8.

```

για κ από 1 μέχρι 4
  διάβασε A[κ * 2]
τέλος_επανάληψης
για κ από 1 μέχρι 4
  A[2 * κ - 1] ← 0
τέλος_επανάληψης

```

17.36 Να συμπληρώσετε τα στοιχεία του πίνακα Γ:

Γ[1]	Γ[2]	Γ[3]	Γ[4]	Γ[5]	Γ[6]	Γ[7]	Γ[8]

μετά την εκτέλεση του ακόλουθου τμήματος αλγόριθμου:

για κ από 1 μέχρι 8
 αν κ mod 2 = 0 τότε
 $\Gamma[\kappa] \leftarrow \Gamma[\kappa - 1] * 2$
 αλλιώς
 $\Gamma[\kappa] \leftarrow \kappa * 2$
 τέλος_αν
 τέλος_επανάληψης

17.37 Δίνεται μονοδιάστατος πίνακας A, 5 θέσεων, ο οποίος στις θέσεις 1 έως 5 περιέχει αντίστοιχα τους αριθμούς 12, 4, -1, 7, 10, καθώς και τμήμα αλγόριθμου:

για κ από 2 μέχρι 4
 $A[\kappa] \leftarrow A[\kappa - 1] + A[\kappa + 1]$
 τέλος_επανάληψης

Να γράψετε τι τιμή θα έχει κάθε στοιχείο του πίνακα μετά την εκτέλεση του παραπάνω τμήματος.

17.38 Δίνεται πίνακας A, 10 θέσεων, ο οποίος στις θέσεις 1 έως 10 περιέχει αντίστοιχα τους αριθμούς 15, 3, 0, 5, 16, 2, 17, 8, 19, 1, καθώς και τμήμα αλγόριθμου:

Για i από 1 μέχρι 9 με_βήμα 2
 $k \leftarrow ((i + 10) \bmod 10) + 1$
 $A[i] \leftarrow A[k]$
 Εκτύπωσε i, k, A[i], A[k]
 Τέλος_επανάληψης

Ποιες τιμές τυπώνονται με την εντολή Εκτύπωσε i, k, A[i], A[k] καθώς εκτελείται το παραπάνω τμήμα αλγόριθμου; (Εξετάσεις 2002)

17.39 Δίνεται ο μονοδιάστατος πίνακας C με έξι στοιχεία, που έχουν αντίστοιχα τις παρακάτω τιμές 2, 5, 15, -1, 32, 14, και το παρακάτω τμήμα αλγόριθμου:

min ← 100
 max ← -100
 για i από 1 μέχρι 6 με_βήμα 2
 $A \leftarrow C[i]$
 $B \leftarrow C[i + 1]$
 αν A < B τότε
 $L_{\min} \leftarrow A$
 $L_{\max} \leftarrow B$
 αλλιώς
 $L_{\min} \leftarrow B$
 $L_{\max} \leftarrow A$

τέλος_αν
 αν $L_{min} < \min$ τότε
 $\min \leftarrow L_{min}$
 τέλος_αν
 αν $L_{max} > \max$ τότε
 $\max \leftarrow L_{max}$
 τέλος_αν
 εκτύπωσε A, B, L_{min} , L_{max} , \min , \max
 τέλος_επανάληψης
 $D \leftarrow \min * \max$
 εκτύπωσε D

Να εκτελέσετε το παραπάνω τμήμα αλγόριθμου και να γράψετε στο τετράδιό σας:
 i. τις τιμές των μεταβλητών A, B, L_{min} , L_{max} , \min και \max , όπως αυτές εκτυπώνονται σε κάθε επανάληψη.
 ii. την τιμή της μεταβλητής D που εκτυπώνεται. (Εξετάσεις 2004)

17.40 Έστω τρεις πίνακες A, B και Γ, 5 θέσεων, με τους δύο πρώτους να περιέχουν τις ακόλουθες τιμές:

A[1]	A[2]	A[3]	A[4]	A[5]
5	6	2	0	3

B[1]	B[2]	B[3]	B[4]	B[5]
0	7	2	1	-3

Να γράψετε τι τιμές θα έχει ο πίνακας Γ, αν εκτελεστεί το ακόλουθο τμήμα αλγόριθμου:

για i από 1 μέχρι 5
 αν $B[i] > A[i]$ και $i \bmod 2 = 0$ τότε
 $\Gamma[i] \leftarrow B[i]$
 αλλιώς
 $\Gamma[i] \leftarrow A[i]$
 τέλος_αν
 τέλος_επανάληψης

17.41 Δίνεται ο ακόλουθος μονοδιάστατος πίνακας T με 8 στοιχεία που περιέχει τις ακόλουθες τιμές:

	1	2	3	4	5	6	7	8
T	1	1	2	1	4	1	1	4

καθώς και ο πίνακας Γ, 5 θέσεων με όλα του τα στοιχεία μηδενικά δηλαδή:

	1	2	3	4	5
Γ	0	0	0	0	0

Να γράψετε τι τιμές θα περιέχει ο Γ, αν εκτελεστεί το ακόλουθο τμήμα αλγόριθμου:

για θ από 1 μέχρι 8
 $\Gamma[T[\theta]] \leftarrow \Gamma[T[\theta]] + 1$
τέλος_επανάληψης

17.42 Τέσσερις μαθητές πρότειναν τα επόμενα τμήματα αλγόριθμων για τον υπολογισμό του αθροίσματος 100 αριθμών εισόδου. Να χαρακτηρίσετε κάθε πρόταση μαθητή ως σωστή ή λάθος.

Μαθητής 1

$\Sigma \leftarrow 0$
για κ από 1 μέχρι 100
διάβασε x
 $\Sigma \leftarrow \Sigma + x$
τέλος_επανάληψης

Μαθητής 3

$\Sigma \leftarrow 0$
για κ από 1 μέχρι 100
διάβασε x
τέλος_επανάληψης
για κ από 1 μέχρι 100
 $\Sigma \leftarrow \Sigma + x$
τέλος_επανάληψης

Μαθητής 2

$\Sigma \leftarrow 0$
για κ από 1 μέχρι 100
διάβασε $A[\kappa]$
 $\Sigma \leftarrow \Sigma + A[\kappa]$
τέλος_επανάληψης

Μαθητής 4

$\Sigma \leftarrow 0$
για κ από 1 μέχρι 100
διάβασε $A[\kappa]$
τέλος_επανάληψης
για κ από 1 μέχρι 100
 $\Sigma \leftarrow \Sigma + A[\kappa]$
τέλος_επανάληψης

17.43 Τέσσερις μαθητές πρότειναν τα επόμενα τμήματα αλγόριθμων για την εμφάνιση 100 αριθμών με αντίθετη σειρά από αυτήν που εισήχθησαν. Να χαρακτηρίσετε κάθε πρόταση μαθητή ως σωστή ή λάθος.

Μαθητής 1

για κ από 1 μέχρι 100
διάβασε x
τέλος_επανάληψης
για κ από 100 μέχρι 1 με_βήμα -1
εμφάνισε x
τέλος_επανάληψης

Μαθητής 3

για κ από 1 μέχρι 100
διάβασε $A[\kappa]$
τέλος_επανάληψης
για κ από 1 μέχρι 100
εμφάνισε $A[101 - \kappa]$
τέλος_επανάληψης

Μαθητής 2

για κ από 1 μέχρι 100
διάβασε $A[\kappa]$
τέλος_επανάληψης
για κ από 100 μέχρι 1 με_βήμα -1
εμφάνισε $A[\kappa]$
τέλος_επανάληψης

Μαθητής 4

για κ από 1 μέχρι 100
διάβασε $A[\kappa]$
εμφάνισε $A[101 - \kappa]$
τέλος_επανάληψης

17.44 Τέσσερις μαθητές στην άσκηση να γίνει αλγόριθμος ο οποίος να διαβάζει 50 αριθμούς και να τους καταχωρίζει σε πίνακα 50 θέσεων κάνοντας παράλληλα έλεγχο εγκυρότητας ώστε να ανήκουν στο διάστημα $[0, 100]$, έδωσαν τα παρακάτω τμήματα αλγόριθμου. Να χαρακτηρίσετε κάθε τμήμα ως σωστό ή λάθος.

Μαθητής 1

για κ από 1 μέχρι 100
διάβασε x
όσο $x < 0$ ή $x > 100$ επανάλαβε
 διάβασε x
 τέλος_επανάληψης
Α[κ] ← x
τέλος_επανάληψης

Μαθητής 3

για κ από 1 μέχρι 100
διάβασε Α[κ]
όσο $A[k] < 0$ ή $A[k] > 100$ επανάλαβε
 διάβασε Α[κ]
 τέλος_επανάληψης
τέλος_επανάληψης

Μαθητής 2

για κ από 1 μέχρι 100
αρχή_επανάληψης
 διάβασε x
 μέχρις_ότου $x \Rightarrow 0$ και $x \leq 100$
 τέλος_επανάληψης
Α[κ] ← x

Μαθητής 4

για κ από 1 μέχρι 100
αρχή_επανάληψης
 διάβασε Α[κ]
 μέχρις_ότου $\kappa \Rightarrow 0$ και $\kappa \leq 100$
 τέλος_επανάληψης

17.45 Να γράψετε αλγόριθμο που θα δέχεται τα στοιχεία ενός μονοδιάστατου πίνακα 100 στοιχείων και θα εμφανίζει το διπλάσιο κάθε στοιχείου του.

17.46 Να γράψετε αλγόριθμο που θα διαβάζει 100 ακέραιους αριθμούς και θα εμφανίζει το διπλάσιο όλων των στοιχείων του, αλλά με αντίθετη φορά από αυτή που διαβάστηκαν.

17.47 Να γράψετε αλγόριθμο ο οποίος θα διαβάζει 100 ακέραιους αριθμούς και θα τους καταχωρίζει σε πίνακα 100 θέσεων. Να εμφανίζει τα στοιχεία του πίνακα που βρίσκονται σε άρτιες θέσεις και είναι περιττοί αριθμοί.

17.48 Να γράψετε αλγόριθμο που θα διαβάζει έναν πίνακα 100 θέσεων με ακέραιους αριθμούς και θα εμφανίζει το άθροισμα των στοιχείων που βρίσκονται σε περιττές θέσεις και συγχρόνως διαιρούνται με το 4.

17.49 Να γράψετε πρόγραμμα το οποίο θα διαβάζει 30 τριψήφιους θετικούς ακέραιους αριθμούς και θα τους καταχωρίζει σε πίνακα 30 θέσεων. Το πρόγραμμα θα εμφανίζει τον μέσο όρο αυτών που αποθηκεύτηκαν σε περιττές θέσεις και τουλάχιστον ένα από τα ψηφία τους είναι το 8. Θεωρήστε ότι ο χρήστης θα δώσει έγκυρες τιμές.

17.50 Να γράψετε αλγόριθμο ο οποίος θα διαβάζει 100 διψήφιους θετικούς ακέραιους αριθμούς και θα τους καταχωρίζει σε πίνακα κάνοντας παράλληλα έλεγχο εγκυρότητας ότι οι αριθμοί είναι θετικοί διψήφιοι (έλεγχος για το αν είναι ακέραιοι δεν απαιτείται). Να εμφανίζει τον μέσο όρο αυτών που καταχωρίστηκαν σε άρτιες θέσεις και παράλληλα το πρώτο τους ψηφίο είναι μικρότερο του δεύτερου.

17.51 Δεδομένου ενός πίνακα N θέσεων με πραγματικούς αριθμούς, να γράψετε αλγόριθμο που θα εμφανίζει τις θέσεις του πίνακα στις οποίες είναι αποθηκευμένος αριθμός μεγαλύτερος του 50.

17.52 Δεδομένου ενός πίνακα A , N θέσεων, να γράψετε αλγόριθμο ο οποίος θα δημιουργεί έναν πίνακα B , N θέσεων, του οποίου το κάθε στοιχείο του θα είναι το γινόμενο της θέσης στοιχείου με το αντίστοιχο στοιχείο του πίνακα A .

17.53 Να γράψετε αλγόριθμο ο οποίος θα δημιουργεί τον ακόλουθο πίνακα, 50 θέσεων.

	1	2	3	4	...	50
A	-10	-8	-6	-4	...	88

17.54 Να γράψετε αλγόριθμο ο οποίος θα δημιουργεί τον ακόλουθο πίνακα, 100 θέσεων.

	1	2	3	4	...	100
A	5	2	-4	-16	...	$A[99] * 2 - 8$

17.55 Να γράψετε αλγόριθμο ο οποίος θα δημιουργεί τον ακόλουθο πίνακα, 50 θέσεων.

	1	2	3	4	...	100
A	2	6	4	-2	...	$A[99] - A[98]$

17.56 Να γράψετε αλγόριθμο ο οποίος θα διαβάζει 100 ακέραιους αριθμούς και θα τους αποθηκεύει σε πίνακα 100 θέσεων. Ο αλγόριθμος θα εμφανίζει τον μέσο όρο των στοιχείων του πίνακα που βρίσκονται σε άρτιες θέσεις και συγχρόνως το στοιχείο που βρίσκεται στην επόμενη θέση είναι περιττός αριθμός.

17.57 Να γράψετε αλγόριθμο ο οποίος θα διαβάζει δύο πίνακες, 10 θέσεων, και θα δημιουργεί έναν νέο πίνακα, 20 θέσεων, που στις πρώτες 10 θέσεις θα περιέχει τα στοιχεία του πρώτου πίνακα και στις επόμενες 10 τα στοιχεία του δεύτερου πίνακα.

17.58 Να γράψετε αλγόριθμο που θα διαβάσει δύο ακέραιους πίνακες A και B 20 θέσεων και θα δημιουργεί έναν πίνακα Γ, 20 θέσεων, στον οποίο σε κάθε θέση του θα αποθηκεύεται το μεγαλύτερο από τα στοιχεία των A και B της αντίστοιχης θέσης.

17.59 Να γράψετε αλγόριθμο ο οποίος θα διαβάσει έναν πίνακα A, 100 θέσεων, και θα δημιουργεί έναν άλλον πίνακα B, 99 θέσεων, που το κάθε στοιχείο του θα αποτελεί το άθροισμα των στοιχείων του πίνακα A που βρίσκονται στην αντίστοιχη και στην επόμενη θέση του.

17.60 Να γράψετε αλγόριθμο ο οποίος θα διαβάσει 100 αριθμούς και θα τους αποθηκεύει σε έναν πίνακα A, 100 θέσεων. Στη συνέχεια θα δημιουργεί δύο πίνακες B και Γ, 20 και 80 θέσεων αντίστοιχα, όπου στον πίνακα B θα καταχωριστούν τα 20 πρώτα στοιχεία του πίνακα A και στον πίνακα Γ τα επόμενα 80.

17.61 Να γράψετε αλγόριθμο ο οποίος θα δέχεται από τον χρήστη το πλήθος των λέξεων μιας πρότασης, το οποίο δεν πρέπει να ξεπερνάει το 1.000. Στη συνέχεια θα δέχεται τις λέξεις αυτές και θα τις αποθηκεύει σε πίνακα 1.000 θέσεων. Τέλος θα εμφανίζει τις λέξεις του πίνακα ξεκινώντας από την τελευταία παρεμβάλλοντας ένα κενό (“ ”) μεταξύ τους. **Σημείωση:** Για το πλήθος των λέξεων να γίνεται έλεγχος εγκυρότητας.

17.62 Να γράψετε αλγόριθμο ο οποίος για δεδομένο έναν πίνακα N θέσεων θα διαβάσει έναν ακέραιο αριθμό που θα ανήκει στο κλειστό διάστημα $[1, N]$ και δημιουργεί έναν καινούριο πίνακα $N - 1$ θέσεων ο οποίος θα περιέχει τα στοιχεία του πρώτου πίνακα εκτός του στοιχείου που η θέση του καθορίζεται από τον ακέραιο αριθμό που έδωσε ο χρήστης.